Adaptive Bio-Inspired Wireless Network Routing for Planetary Surface Exploration

Richard L. Alena

NASA Ames Research Center
Moffet Field, CA. 94035
650-604-0262
Richard.L.Alena@nasa.gov

Charles Lee
SAIC
NASA Ames Research Center
Moffet Field, CA. 94035
650-604-5054
clee@mail.arc.nasa.gov

Abstract—Wireless mobile networks[1][2] suffer connectivity loss when used in a terrain that has hills and valleys when line of sight is interrupted or range is exceeded. To resolve this problem and achieve acceptable network performance, we have designed an adaptive, configurable, hybrid system to automatically route network packets along the best path between multiple geographically dispersed modules. This is very useful in planetary surface exploration, especially for ad-hoc mobile networks, where computational devices take an active part in creating a network infrastructure, and can actually be used to route data dynamically and even store data for later transmission between networks. Using inspiration from biological systems, this research proposes to use ant trail algorithms with multi-layered information maps (topographic maps, RF coverage maps) to determine the best route through ad-hoc networks at real time. The determination of best route is a complex one, and requires research into the appropriate metrics, best method to identify the best path, optimizing traffic capacity, network performance, reliability, processing capabilities and cost. Real ants are capable of finding the shortest path from their nest to a food source without visual sensing through the use of pheromones. They are also able to adapt to changes in the environment using subtle clues. To use ant trail algorithms, we need to define the probability function. The artificial ant is, in this case, a software agent that moves from node to node on a network graph. The function to calculate the fitness (evaluate the better path) includes: length of the network edge, the coverage index, topology graph index, and pheromone trail left behind by other ant agents. Each agent modifies the environment in two different ways:

Local trail updating: As the ant moves between nodes it updates the amount of pheromone on the edge.

Global trail updating: When all ants have completed a tour the ant that found the shortest route updates the edges in its path

The purpose of the local updating is mainly to avoid very strong pheromone edges to be chosen by every ant and hence to increase exploration and hopefully avoid locally optimal solutions. The global updating function gives the shortest path higher reinforcement, which is a higher amount of pheromone on the edges of the path.

In addition the agents are provided with some capabilities not present in real ants, but likely to help solving the problem at hand. For example each ant is able to determine how far away nodes are, what the RF coverage index is, topology favorable index and they all have a memory of which nodes they have already visited. Furthermore, we add the estimated values for next node by tracking the speed of current mobile units. The simulation shows that the method is feasible and more reliable. It is a feasible way to avoid node congestion and network interruptions without much decrease of network performance.

TABLE OF CONTENTS

1. INTRODUCTION 1
2. ROUTING PROTOCOL 2
3. BIO-INSPIRED ANT TRAIL ALGORITHMS 3
4. EXPERIMENTS AND SIMULATIONS 4
5. CONCLUSION 5
7. FUTURE DEVELOPMENT 5
REFERENCES 5
BIOGRAPHY 6

1. INTRODUCTION

In planetary surface exploration, humans and robots are all moving objects in the field. Since they all carry computation capability and are connected to each other, they are moving wireless network nodes. The robots are usually equipped with a computer as a part of their components and humans are equipped with a backpack that has computers in it. The motion feature and the tough terrain surface of the exploration site makes the task hard. The surface that we are exploring has hills and valleys. When the network nodes are moving, they may go beyond line of sight, they may be in the location that has multi-path interference, they may be in a place where radio signals are absorbed. All those phenomenon become the road blocks of wireless network construction. Not only do they degrade the performance of

[2] IEEEAC paper #1584, Version 5, Updated Dec 17, 2004

the wireless network, but they also cause interruption of the connections. In our tasks of planetary surface explorations, many methods are utilized and each solves certain problems [1,2]. In this paper, we present a new method to solve the performance and interruption problems of such networks. We have implemented the simulation software to get approximate results. We know that there are existing routing algorithms in commercial products, but most of them are not designed for our situation. Some proactive routing schema, like Destination Sequenced Distance Vector (DSDV) [3], continuously update the routing tables of mobile nodes consuming a large portion of the network capacity for huge routing table data. This reduced the available capacity of the network for actual data communication. The on demand routing protocols like on-demand Ad Hoc On-demand Distance Vector and Dynamic Source routing (AODV) [4,5], on the other hand, uses routing discovery, and require the actual communication to be delayed until the route is determined. This may not be suitable for our real time data and multimedia communication that require bandwidth and speed. Our exploration requires that sensor readings be broadcasted to the nodes in real time and require that our network has bandwidth of transmitting images and sometimes videos. Inspired by ant trail phenomena, ant algorithms are proposed by Gianni Di Caro [3]. The algorithms can be used in the network routing by adaptively learning the best paths and increase performance of the networks. However, when the network topology has dynamic features, pure ant based routing couldn't perform well since the nodes can not start communication until the ant discovers the best path and provides the routes, which will keep the data stored in the buffer until the ants come back with optimal paths. In other cases, if the nodes that carry ants suddenly disconnect from rest of network due to moving out of the RF coverage area or just because of multi-path interference, the waiting time could be infinite. Marwaha, et. al. proposed a hybrid method to solve such problems by combining AODV and ant (ant-AODV) [2]. This method is able to reduce the end-to-end delay, but still added more overhead to the network and didn't resolve the problems such as in the situation when the mobile nodes move out of RF coverage. In another words, it possesses the same shortcoming as AODV has in such cases. We propose the method that uses ant algorithms with RF mapping using GPS information. We can calculate the moving speed of nodes and predict the future point of the nodes to forecast future connectivity. This reduced the chances that ants are cut out of the network and sudden network interruptions caused by waiting for the ants to discover routes. We do also add some overhead to the basic ant algorithms with those calculations, but it only costs some CPU time so it is not as significant. We have compared our results with others and the performance is better in the situation that some nodes move out of RF coverage. Our methods ensure network connectivity and performance even with some nodes cut out. Because we only update the whole routing table when there is a prediction that some nodes may be out of the network, the portion of network capability used by updating messages is not as high as AODV. The simulation shows that the method can achieve a better end-to-end delay time in comparison with the AODV, and Hybrid algorithm presented by Marwaha, et al.

2. ROUTING PROTOCOL

There are different routing methods in wireless networks; dynamic and static. Both methods route the packets by using routing tables. In dynamic routing, rather than computing the entire path to a destination, it merely selects the next hop leading to that destination, and relies on the next hop machine to select a further hop that gets the packet closer to its destination. Independent hop-by-hop routing requires that all machines have a consistent view of how to reach all destinations in the network. If consistency is lost, two or more machines (presumably routers) can form a routing loop, and the packet never makes it to its destination.

To achieve consistency, a network administrator can either manually configure each machine with a pre-computed set of routes that he or she knows to be consistent, or the machines can communicate routing information to each other through some kind of protocol. The first approach is known as static routing, and the second as dynamic routing.

Static routing has some enormous advantages over dynamic routing. Chief among these advantages is predictability. Because the network administrator computes the routing table in advance, the path a packet takes between two destinations is always known precisely, and can be controlled exactly. With dynamic routing, the path taken depends on which devices and links are functioning, and how the routers have interpreted the updates from other routers.

Additionally, because no dynamic routing protocol is needed, static routing doesn't impose any overhead on the routers or the network links. While this overhead may be minimal on an FDDI ring, or even on an Ethernet segment, it could be a significant portion of network bandwidth on a low-speed link. Consider a network with 200 network segments. Every 30 seconds, as required by the RIP specification, the routers all send an update containing reachability information for all 200 of these segments. With each route taking 16 octets of space, plus a small amount of overhead, the minimum size for an update in this network is over three kilobytes. Each router must therefore send a 3 Kb update on each of its interfaces every 30 seconds. As you can see, for a large network, the bandwidth devoted to routing updates can add up quickly.

The chief advantages of dynamic routing over static routing are scalability and adaptability. A dynamically routed network can grow more quickly and larger, and is able to adapt to changes in the network topology brought about by this growth or by the failure of one or more network components.

With a dynamic routing protocol, routers learn about the network topology by communicating with other routers. Each router announces its presence, and the routes it has available to the other routers on the network. Therefore, if you add a new route, or add an additional segment to an existing router, the other routers will hear about the addition and adjust their routing tables accordingly. You don't have to reconfigure the routers to tell them that the network has changed. Similarly, if you move a network segment, the other routers will hear about the change. You only need to change the configuration of the router (or routers) that connect the segment that moved. This reduces the chance that errors will occur.

The ability to learn about changes to the network's configuration has implications beyond adding new segments or moving old ones. It also means that the network can adjust to failures. If a network has redundant paths, then a partial network failure appears to the routers as if some segments got moved (they are now reached via alternate paths), and some segments have been removed from the network (they are now unreachable). In short, there's no real difference between a network failure and a configuration change. Dynamic routing allows the network to continue functioning, perhaps in a degraded fashion, when a partial failure occurs, provided there are redundant paths.

Static routing has higher performance in comparison with dynamic routing. We will use dynamic routing in our experiment and simulation even though the dynamic routing has the disadvantage of degraded performance. Since we know that static routing is not suitable for the difficultenvironment where connection loss is the main concern, static routing will not ensure connectivity for moving nodes and ad hoc networks.

Dynamic routing needs to calculate the routing paths. This is a very time-consuming task if we use the traditional ways to calculate the cost of all the paths and select the best one. Their calculation could be an NP-complete problem. And furthermore, to pass the routing tables to the network will cost capacity of the network. In addition, some other overhead will be added if we consider balancing the network traffic for the routing. When the ant trail algorithm is used, we use small data grams traveling on the network to find the best route instead of calculating it, plus the parallel feature of distributed processing, the decrease on the calculation cost is substantial.

We adopt the ad hoc vector routing protocol proposed by Nokia; dealing with route table management. Route table information must be kept even for short-lived routes, such as are created to temporarily store reverse paths towards nodes originating RREQs. AODV uses the following fields with each route table entry:

- Destination IP Address
- Destination Sequence Number
- Valid Destination Sequence Number flag
- Other state and routing flags (e.g., valid, invalid, repairable, being repaired)
- Network Interface
- Hop Count (number of hops needed to reach destination)
- Next Hop
- List of Precursors
- Lifetime (expiration or deletion time of the route)

3. Bio-inspired Ant Trail Algorithms

We use the phenomenon of ant trails when ants are looking for food to improve our network routing. Since ants can always find the best path in this way, we will program some "ants" in the network to find the best path for us. The method of ant trail network routing include three steps:

- Choosing next node to travel to
- Updating trip time list
- Calculate changes in the network.

We will need to update the routing table to keep the information up-to-date. The reason is that the nodes are all moving and the routing table should be changing along with the moving of nodes. Some nodes may be just out of the signal coverage area, some nodes may be in the less favorable area where multiple paths, echo, weak signal, and noise may play a role to decrease the reliability of network connectivity. However, as we had mentioned earlier; the information updating has a cost to the network capacity, we need to be careful on how often to update the network routing table. In our experiments, we make frequency of the updates a variable so that we can set it to different values in the field tests. The updating consists of updating next node selection, updating the node RF coverage and favorable propagation situation, updating the topography map, and the RF coverage map, some of those are local updates without having any impact on the network traffic. We need to consider updating the amount of pheromone on the edge each time the ants move through it.

To evaluate the goodness of each path, we need to quantify the value of the path by computing th traveling time through it. In the computer software world, they are list and stack; the stack stores the history of the trip time and the list stores the updated trip time, we also call it trail metrics. In another words, the ants are the small datagrams moving in the network to pioneer the route and find out the cost. The procedures for ant's activities are as follows:

1. Every node sends out a "forward ant" to a randomly selected destination at regular intervals. The ant maintains a history-stack of nodes visited enroute as well as the elapsed times.

2. Ants select the next hop using probability information in the routing tables. If a chosen next node has already been visited, a uniformly random selection among the neighbors is applied. A tiny exploration probability allows ants to choose the next hop with even probability given to all neighbor nodes.

3. Cycles or loops are removed from the history-stack when detected. The predicted unreachable nodes are removed from the history stack and from the network so they will not be counted when ants go for the next trail.

4. When the destination node is reached, a "backward ant" is generated with the same memory as the "forward ant". The "backward ant" follows the found path in the opposite direction by popping nodes from the history-stack.

5. At each node k coming from node f, the backward ant updates the routing table by increasing the probability of $P(d,f)$ associated with node f when the destination is d and decreasing $P(d,n)$ of the other neighbors. The ant also updates a list Trip_k with values taken from the history-stack. Trip elapsed times from node k to all nodes visited before node k on the forward ant's path is used to update the corresponding sample means and variances.

Trip times are used as an indication of goodness of a taken path because it is proportional to the number of hops, link capacities and processing speed of nodes crossed on the way. Important to note is that forward ants have the same queuing priority as data, but that backward ants have higher priority to propagate accumulated information faster. Links on congested paths will be given only a little, very delayed reward. When the ants pick the path to travel, they pick the link with the most pheromone deposit as priority route. And the pheromone is updated after they go through the link.

The goodness and stability of an observed trip time T is evaluated on the basis of the estimated mean and variance values stored in the Trip list. This is used to compute by how much the routing table should be adjusted. The implementation of the ant trail algorithms can be summarized in the following steps.

1. {Initialization}
   Initialize trip_history and Trip_time, and
   topography RF coverage map.
   {Construction}
   For each ant k (currently in state i) do
   repeat
   choose, in probability, next hop to move into.
   append the chosen move to the k-th ant's set table.
   until ant k has completed its solution.
   end for
2. {Trail update}
   For each ant move do
   Compute the cost
   update the trail matrix Trip_time, push history.

   end for
3. {Terminating condition}
   If not(end test) go to step 2

4. Experiments and Simulations

In order to understand whether this method improves the network performance, we conducted an experiment. We used common simulation environments so that comparison of results with others will be easy. We used the NS-2 as a network simulator to simulate the protocols. Popular CSMA/CA is used to transmit these packets. The simulated network has 50 mobile nodes within an area of 1500m by 300m with speed of 0-10 m/s. We apply a topography map on the top of the coverage area. The RF coverage and the signal reachable index are preset in each point of the area with a resolution of one meter. The simulation was run multiple times with 6 different pauses. Each pause time 0, 30, 120, 300 and 600 seconds. The purpose of the pause time is to reset the source and destination nodes so that the network is not always transmitting packets to and from the same sets of nodes. Also, the pause time can clear the buffer for the next run. After each pause, the new destination is selected and speed is re-established between 0 and 10 m/s. Source nodes and destination nodes are chosen at random in uniform distribution. When the nodes move, we track it by its coordinate, represented as (x, y). Since those coordinates are very similar to GPS coordinates, instead of Longitude and Latitude they are x and y, we can simulate the GPS-driven RF coverage map. When a node moves to the coordinate where there is no RF coverage, we set the node as unreachable. By using the GPS information; coordinate and speed of a node, we can calculate and predict when the nodes will be unreachable. When a node is unreachable, we will update the routing table globally. Because we don’t need ants to detect the unreachable situation, the network performance will be greatly improved. Therefore, location predictions prevent unreachable detection by ants, the most time consuming task.

We have distinguished the global and local update, the local updates doesn’t cost the capacity of the network, global updates cost the capacity of network. We limited the global updates by ants algorithms in comparison with the AODV algorithms. We only use AODV in the very limited way so that we reduced the network traffic and updating time delays.

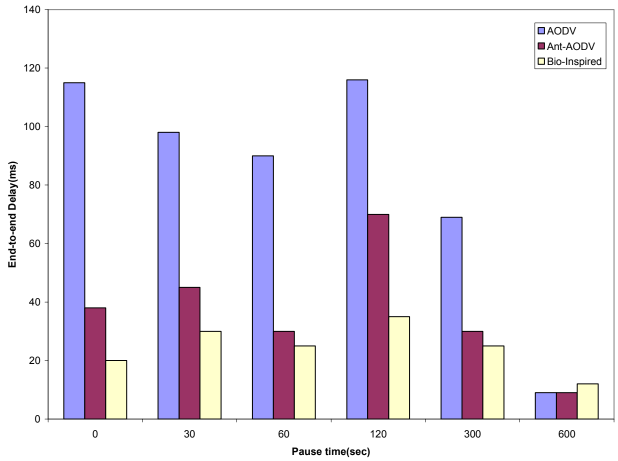

Figure 1: End-to-end Delay

Figure 1 shows the experiment results for end-to-end delay. The x axis is the different pause time and the y axis is the end-to-end delay time in seconds. We can see that the our method, Bio method in chart, is the least delay for the experiment. The end-to-end delay reflects how much time the packet travels from one node to the other and shows the performance of the network.

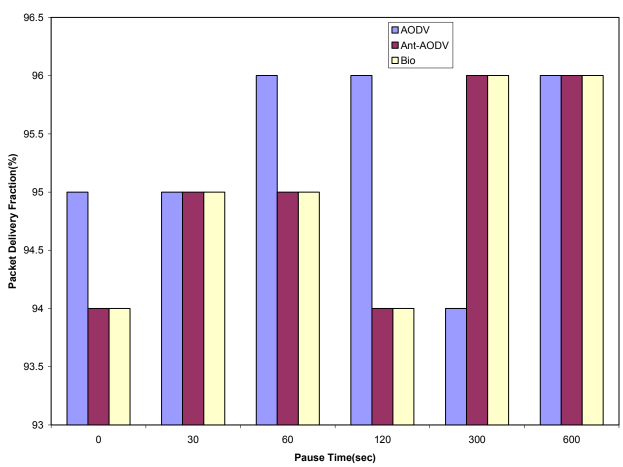

Figure 2: Packets transmission

Figure 2 shows the packet transmission effectiveness. We can see that the Bio-inspired methods have the best performance in comparison with the AODV and the Hybrid method by Marwaha, et al.

5. CONCLUSION

We have presented a new method to route wireless network packets to increase network performance. The experiments show that the method reduced the end-to-end delay by a noticeable amount of time. The feature of our method is to use the combination of Ant trail algorithms and AODV routing algorithms with minimum global routing table updates. We added the artificial intelligence by predicting the unreachable nodes by calculating the next location of node movement and remove them out of network virtually. With the help of topography map, we can predetermine the location that is not covered by the RF signal. The comparison charts have been drawn to compare the method with some other improved wireless network and we have shown that by simulation results.

7. FUTURE DEVELOPMENT

This method has shown the increase of the network performance by decreasing the end-to-end delay. While this bio-inspired network has laid important foundations for the performance test, bandwidth performance experiments could be a further research topic. There are more aspects that we can do research on to improve bandwidth performance. One is through global optimization. When ant travels the network to look for the optimized route, we can see that it could ignore the better path in some conditions because it sees the more pheromones on it's neighbor link and not look further for better link from global point of view. If this "short sighted" view become the basis to make decision, which lead to unbalanced network traffic, bandwidth performance is degraded. To overcome this shortcoming, we could introduce a global view of traffic balance factor so the ant could have a global "view" when making decisions for the next route. This controversial topic is more complex but can be done in future research and development.

REFERENCES

[1] Richard L. Alena, John Ossenfort, Charles Lee, Edward Walker, Thom Stone, "Design of Hybrid Mobile Communication Networks for Planetary Exploration," Big Sky, MT, February 2004.

[2] Gianni di Caro, Marci Dorigo, AntNet: Sistributed Stigmergetic Control for Communications Networks" in Kournal of Artificial Intelligence

Research 9 pp 317-365 December, 1999.

[3] Lu Guoying, Liu Zemin, and Zhou Zheng "Multicast routing based on ant algorithm for delay-bounced and lad-balancing traffic." 25th Annual IEEE Conference on Local Networdks, Nov 8 – 10, 2000, Tempa, FL

[4] C. E. Perkins, E. M. Royer and S. R. Das, "Ad Hoc On-demand Distance Vector (AODV) Routing" in Proceeding of IEEE Workshop on Mobile Computing Systems and Applications, pp. 90-100, Feburaru 1999

[5] R.R. Choudhary, S. Bhandhopadhyay and K. Paul, "A Distributed Mechanism for topology discovery in ad Hoc Wireless Networks Using Mobile Agents", in Proceedings of Modicom, pp. 145-146, August, 2000

[6] Ping. Li, Richard. E. Haskell, Darrin. M. Hanna, "Optimizing Fuzzy Decision Tree by Using Genetic Algorithms," Proceedings of the International Conference on Artificial Intelligence, June, 2003, Las Vegas, USA

BIOGRAPHY

Richard L. Alena *is a Computer Engineer and Group Lead for the Intelligent Mobile Technologies research and development team in Computational Sciences Division at NASA Ames. He led the design and development of distributed mobile data systems supporting geological and biological scientific surveys in the Canadian Arctic and American Desert, investigating advanced computing solutions for planetary exploration and coordinating satellite and wireless networks with wearable computing for multi-agent simulations. Mr. Alena is the co-lead for the joint ARC-JSC Advanced Diagnostic Systems for International Space Station Project, developing model-based diagnostic tools for space operations. He is also involved with TEAMs and Livingstone researchers, supporting development of subsystem interaction models and Caution and Warning analysis. As a senior computer scientist he was the chief architect of a flight experiment conducted aboard Shuttle and Mir using laptop computers, personal digital assistants and servers in a wireless network for the International Space Station. He is also the technical lead for the Databus Analysis Tool for International Space Station on-orbit diagnosis. Mr. Alena holds a B. S. and M.S. in Electrical Engineering and Computer Science from the University of California, Berkeley and holds a U.S. patent for "Three Electrode Hydroquinone Subcutaneous Equilibrating Tonometer." He is the winner of a NASA Silver Snoopy Award in 2002 and a NASA Space Act Award for A Comprehensive Toolset for Model-Based Health Monitoring and Diagnostics. He has also been awarded a JSC Group Achievement Award in 2000 for his participation in the Cockpit Avionics Upgrade Display/Control Application Requirements Team, a NASA Group Achievement Award in 1998 for his work on the ISS Phase 1 Program Team and a Space Flight Awareness Award in 1997.*

Charles Lee, *employed by SAIC, is the Technical Lead on Mobile Agents project at NASA Ames Research Center. He holds a Ph.D. in systems engineering and computer science from Oakland University, in Rochester, Michigan. Completed research projects includes several systems that have been successfully deployed at the Mars Desert Research Station, providing functions for extending human performance and situational awareness into the planetary exploration domain targeting future Mars exploration. These include robust GPS switchboard on-demand services that provide GPS information with awareness of loss and the ability to regain wireless network connections, and a store and forward architecture to maintain data continuity in the event of network connection loss. In addition, Dr. Lee developed distributed agents that serve sensor information through a publish and subscribe architecture in heterogeneous computer environments, and a mapping and planning system that provides location and orientation of mobile rovers and astronauts on topographic maps for navigation planning and real time monitoring. Other work includes joint development of custom software to provide access to avionics data for Advanced Diagnostics System (ADS) applications, and collection and organization of International Space Station (ISS) data sets by fault scenario, along with liaison with ADS developers and users in the design of data interfaces, user interfaces and tools relevant to ADS on ISS. He developed the first version of Caution and Warning cube visualization software that handles the command and data handling events for fault detection.*